IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 3, 2018, accepted October 24, 2018, date of publication November 9, 2018,
date of current version November 30, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2878813

An Enhanced Deep Extreme Learning Machine for Integrated Modular Avionics Health State Estimation

ZEHAI GAO, CUNBAO MA, ZHIYU SHE, AND XU DONG
School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China

Corresponding author: Zehai Gao (gaozehai@mail.nwpu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61873203 and in part by the Aeronautical Science Foundation of China under Grant 20165553035.

ABSTRACT Integrated modular avionics (IMA) is one of the most advanced systems whose performance deeply impact on the security of civil aircraft. In order to enhance the safety and reliability of aircraft, the health state of the IMA must be estimated accurately. Since IMA is a real-time system, the estimation algorithm should have fast learning speed to satisfy the real-time requirement. In this paper, an enhanced deep extreme learning machine is developed to estimate the health states of IMA. First, the enhanced deep extreme learning machine is built in a novel fashion by using a dropout technique and extreme learning machine autoencoder. Second, multiple-enhanced deep extreme learning machines with different activation functions are employed to estimate the health states, simultaneously. Finally, a synthesis strategy is designed to combine all the results of different enhanced deep extreme learning machines. In such a manner, the robust and accurate estimation results can be obtained. In order to collect the data under different health states, a performance degradation model of IMA is built by the intermittent faults. The proposed method is applied to health state estimation, and the results confirm that the proposed method can present a superior estimation to the conventional methods.

INDEX TERMS Integrated modular avionics, extreme learning machine, health state, intermittent faults.

I. INTRODUCTION

Integrated modular avionics (IMA) is an imperative system for airplanes. An Arinc653 based IMA is controlled by a real-time computer network, on which almost all the avionics functions of different criticalities are hosted [1]. IMA system architecture is the main trend of development in the aviation domain [2]. Since IMA is a highly integrated, partitioned and shared computing platform, all the hosted functions are executed successively, according to partition scheduling. This structure can effectively improve the system's efficiency while reducing resource assignment. The failure of a flight function may spread to other flight functions. Hence, the working performance of IMA is crucial to flight security.

IMA is a core system of aircraft and its integration structure increases system complexity and function correlation, which makes a big challenge for flight security. In order to guarantee safety of aircraft and reduce the huge cost due to regular maintenance, many researches are committed to improve the ability of IMA by virtue of structural optimization and redundancy design [3], [4]. Nevertheless, quite few studies focus on the health state estimation of IMA. The failure of IMA not only greatly affects flight mission, but also causes catastrophic accidents and the loss of human lives. Accordingly, IMA health state must be monitored and estimated accurately.

Health state estimation can be treated as a typical pattern recognition problem. Many researches focus on developing the effective health state estimation methods. Tamilselvan and Wang [5] applied deep belief network based classification method to diagnose aircraft engine health state and electric power transformer health state. Liu *et al.* [6] used principal component analysis and fuzzy C-means to capture features, then, the learned features are fed into support vector machine to implement health state assessment and remaining useful life prediction. Yang *et al.* [7] extracted features from the time domain, frequency domain and time-frequency domain. They developed a health index and neural network for estimating and predicting the health states.

Recently, deep learning is wildly used in many fields due to its excellent feature learning capability [8]. Deep learning employs a hierarchical structure with unsupervised

manner for feature extraction [9]. In order to retain as much information as possible and overcome the drawback of the conventional feature extraction, deep learning can automatically capture the features from the raw data by virtue of its attractive characteristic that deep learning can effectively learn the high-level representations via multiple nonlinear transformations. In view of the advantages, deep learning has not only been applied to many frontier fields, but also evolved many traditional fields, such as prognostics and health management. Liu *et al.* [10] applied Gaussian-Bernoulli deep belief networks for fault detection and diagnosis of electronics-rich analog systems. In [11], stacked denoising autoencoder (SDAE) method was adopted to evaluate the health states for signals containing ambient noise and working condition fluctuations. Li *et al.* [12] developed a deep convolution neural network for remaining useful life estimation.

Despite of the success, deep learning machine still has some drawbacks. The salient disadvantages are that deep learning machine will spend a lot of time training the network and require large memory to store the parameters for adjusting. Since IMA is a real-time system, the estimation method should satisfy the real-time requirement. The disadvantages of the conventional deep learning cannot be permitted on a real-time system. Therefore, a fast and accurate estimation method should be developed. Two major issues for IMA estimation method should be figured out. Firstly, the estimation method prefers to use deep learning manner to learn the high-level feature from the raw data by a direct way, instead of manual manner, for retaining completed information. Secondly, the estimation method should have a fast learning speed to satisfy the real-time requirement.

Extreme learning machine (ELM) was proposed by Huang [13] in 2006 as a fast and effective machine learning algorithm. Different from other single hidden layer feedforward neural networks (SLFN), ELM uses the random feature mapping to generate the hidden node parameters and the least square method to determine the output weights. Such revolution greatly alleviates the burden of parameters adjusting caused by iterative algorithm, and thereby guarantees fast learning speed [14]. Many researchers focus on improving the performance of ELM. Incremental ELM and pruned ELM were developed for amending the number of hidden nodes of basic ELM [15], [16]. The online sequential ELM was presented to satisfy the requirement of online testing [17]. In order to enhance the stability, many studies devoted to selecting the input weights and bias of ELM by optimization algorithms [18]. Al ELM is an excellent classifier, the results are heavily dependent on the extracted features.

Inspired by the characteristics of deep learning and ELM, this paper proposes an enhanced deep extreme learning machine (EDELM) for health state estimation. For the purpose of inheriting the advantages of deep learning and ELM, the EDELM is built in a new fashion. The ELM autoencoder and dropout technique are utilized to enhance the performance of the proposed method. Furthermore, different activation functions are employed as the nonlinear transformation functions for multiple EDELMs to learn the features from the original signals, simultaneously. After that, a synthesis strategy is designed to figure out more stable results by integrating the results of the different EDELMs. Finally, an IMA degradation model is built based on the occurrence of the intermittent faults. The proposed method is used to estimate the health state of IMA.

The rest of the paper is organized as follows. The basic ELM and multiple levels ELM are introduced in section 2. Section 3 describes the proposed algorithm EDELM, illustrates synthesis strategy, and details the general procedure of the proposed method for health state estimation. Section 4 demonstrates the learning performance of EDELM by benchmark dataset. An IMA degradation model is built based on the intermittent faults, and the proposed method is applied to IMA health state estimation. Section 5 summarizes the presented research and the future work.

II. RELATED WORK

This section provides a brief review of the related algorithms. Section 2.1 overviews the general learning process of ELM. Section 2.2 introduces the ELM autoencoder and the multiple levels extreme learning machine.

A. EXTREME LEARNING MACHINE

ELM is an important branch of SLFN, in which the hidden layer needs not to be tuned [19]. It means that the input weights and bias can be initialized randomly, thus the output weight is the only one need to be calculated. Given a training data set with n instances as $S = \{(x_i, y_i) | x_i \in R^N, y_i \in R^M, i = 1, 2, \cdots, n\}$, where $x_i = [x_{i1}, x_{i2}, \cdots, x_{iN}]$ and $y_i = [y_{i1}, y_{i2}, \cdots y_{iM}]$ are the input vector and the target output vector of the ith instance, respectively. N is the number of input attributes, M is the number of classes. The relationship between the input samples and the outputs is defined as follows.

$$y_{it} = \sum_{k}^{L} \beta_{kt} \cdot f\left(\sum_{j=1}^{N} w_{jk} \cdot x_{ij} + b_k\right). \tag{1}$$

where w_{jk} refers to the input weight from the jth input node to the kth hidden neuron; b_k is the bias of the hidden layer; β_{kt} represents the output weight from the kth hidden neuron to the tth output node; L is number of hidden neurons; $f(\cdot)$ is any nonconstant piecewise continuous function which satisfies ELM universal approximation capability theorems [20]. The hidden layer can be written in matrix form as follows.

$$H = \begin{bmatrix} f(w_1 \cdot x_1 + b_1) & \cdots & f(w_L \cdot x_1 + b_L) \\ \vdots & \ddots & \vdots \\ f(w_L \cdot x_n + b_L) & \cdots & f(w_L \cdot x_n + b_L) \end{bmatrix}_{n \times L}. \tag{2}$$

The target and the output weight can be defined as matrix $Y = [y_1, y_2, \cdots, y_n]^T$ and matrix $\beta = [\beta_1, \beta_2, \cdots, \beta_L]^T$,

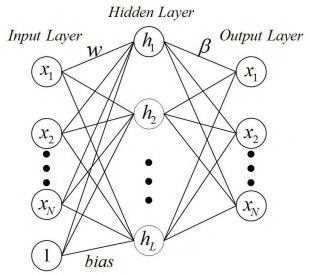


FIGURE 1. The structure of an ELM autoencoder.

respectively. In view of the ELM theory, the goal can be formulated as the following optimization problem:

$$minimize\ J_{ELM} = \frac{\lambda}{2}\|\beta\|^2 + \frac{1}{2}\left\|Y - \hat{Y}\right\|^2. \tag{3}$$

where $\hat{Y}$ is the practical output of ELM; λ is a coefficient of regularization term. According to the KarushKuhnTucker (KKT) theory [21], if the number of instances is larger than the number of hidden neurons, the output weight can be calculated by following equation.

$$\beta = \left(\lambda I + H^T H\right)^{-1} H^T Y. \tag{4}$$

Otherwise, it can be formulated as the following equation.

$$\beta = H^T \left(\lambda I + H H^T\right)^{-1} Y. \tag{5}$$

where I is the identity matrix.

B. MULTIPLE LEVELS EXTREME LEARNING MACHINE

Multiple levels extreme learning machine (ML-ELM) is proposed as a special deep learning, which is constructed by multiple ELM autoencoders [22]. Resembling the conception of basic autoencoder, the target of the ELM autoencoder is set equal to the inputs. The output weight is calculated as the same as basic ELM, but the difference is that the ELM autoencoder adopts the orthogonal random weights and biases to map the input data to a different or equal dimension space. In such manner, ELM autoencoder can have more locality preserving power and a better generalization ability [23]. The structure of ELM autoencoder is shown in Fig. 1.

As a kind of autoencoder, ELM autoencoder firstly projects the inputs to feature space, and then reconstructs the inputs by the analytical output weight β. Multiple ELM autoencoders are stacked one by one to construct ML-ELM. More concretely, the input samples x are fed into an ELM autoencoder to train the first output weight β_1. Then the transposed matric β_1^T is used for building the first hidden layer to capture features. After that, the first hidden layer is used to train the second output weight β_2, the second hidden layer is obtained by β_2^T. In the same vein, the hidden layer H_i of current ELM autoencoder is regarded as the inputs and target of the next ELM autoencoder to train the output weight β_{i+1}, and the hidden layer H_{i+1} is organized by β_{i+1}^T. Finally, the last hidden layer H_n and the actual output target Y are used to determine the last output weight β_{n+1}. The subscript implies the number of hidden layers. Notice that the hierarchical features are captured layer by layer in an unsupervised manner and all the parameters are directly determined, without iterative adjusting.

III. PROPOSED LEARNING ALGORITHM

In this paper, we propose an enhanced deep extreme learning machine for IMA health state estimation. This method consists of three parts: the conception of the proposed EDELM, the synthesis strategy design and the procedure of the proposed method for health state estimation.

A. ENHANCED DEEP EXTREME LEARNING MACHINE

The proposed EDELM is a hierarchical neural network, and the dropout technique is adopted to improve its generalization capability. The theory demonstrated by Huang *et al.* [24] is that the universal approximation capability of ELM cannot be guaranteed without random projection of the inputs. Since the ML-ELM has not fully exploited the merits of ELM theory, the drawback of ML-ELM has been pointed out in [25].

As mentioned above, random projection should be adopted to map the inputs into feature space. However, in the aspect of feature expression, the learned features are unstable resulting from the random projection. In order to figure out this dilemma, many papers apply the optimization method to decide the input weights of ELM. Although such manners can make the learnt feature stable, it will take time to search the input weights, which makes the optimization method not suit the real-time system. In this paper, the ELM autoencoder is mainly used to decide the weights. The stacked ELM autoencoder is used to learn the high-level features. Without iteration, the proposed method can keep the fast learning speed. The structure of EDELM is shown in Fig. 2.

In Fig. 2, the superscript and the subscript of the variables denote the number of hidden layers and the number of neurons in every individual layer. As Fig. 2 shows, the procedure of the proposed method can be broken down into four steps. The following article details these four steps.

In the first step, the outputs are set equal to the inputs. The input weight w^i and bias b^i are randomly generated to map the inputs x or the learned hidden layer $H^l = \left[h_1^l, h_2^l, \cdots, h_k^l\right]^T$ into a random space. Since orthogonal constraint is unreasonable when the number of the input nodes is different from that of the output ones, input weight and bias are not made orthogonal [25]. The output weight β^i can be calculated as the same as basic ELM. In this step, the ELM autoencoder is utilized for determining the output weights which will be used in the second step.

In the second step, the transposed matrix $\left(\beta^i\right)^T$ is used as the input weights to capture the features from a corrupted input vector $\hat{x}$ or a corrupted hidden layer $\hat{H}^l$. Compared with using random initialization manner, the input weight $\left(\beta^i\right)^T$ obtained by using ELM autoencoder in step 1 can help step 2 to learn more stable features. The typical corrup-

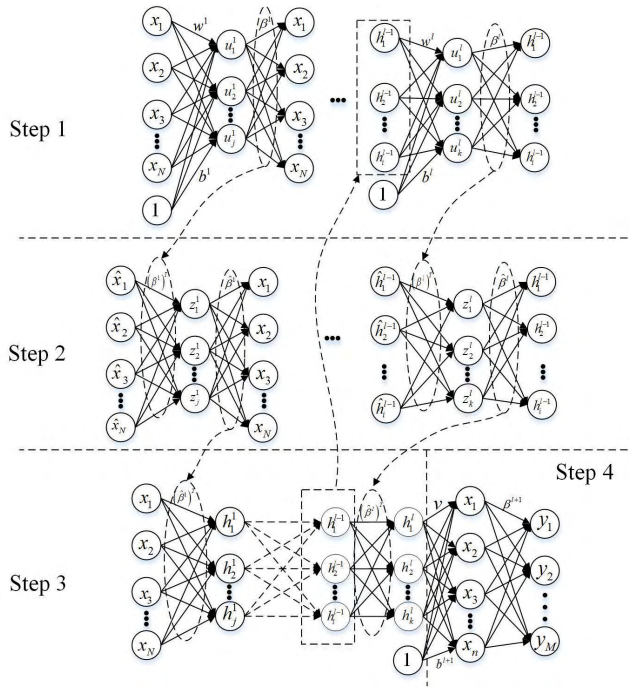


FIGURE 2. The structure of EDELM.

tion methods are masking noise, salt-and-pepper noise and additive Gaussian noise [26]. In this paper, the masking noise is adopted to randomly set the elements of x for each sample to zero under a preset corruption rate. The output weight should reconstruct the clean input x from the corrupted version $\hat{x}$, which can make the learned output weight robust. For further improving the performance, the dropout technique is applied to the hidden layer $Z^l = [z_1^l, z_2^l, \cdots, z_k^l]^T$ in current step. Dropout technique will randomly omit units with a fixed probability for each hidden unit, and then the output weight $\hat{\beta}^i$ should reconstruct the clean input x by using part of the hidden units.

In the third step, the hidden layer $H^l = [h_1^l, h_2^l, \cdots, h_k^l]^T$ is constructed by the transposed matrix $\left(\hat{\beta}^i\right)^T$. The hidden layer H^l is set as the input data and target of the first step. Meanwhile, H^l is also the clean data and target in the second step. The clean input x and learned hidden layers are staked together to construct the deep feature learning machine. The clean input x provide the completed information, besides, the robust weight $\left(\hat{\beta}^i\right)^T$ is working as the link weight, which can further improve the generalization performance of the network and enable the proposed method to attain a good capability for classification.

In the fourth step, the outputs of the third step is regarded as the input features, and the actual output Y is the final target. A random input weight v and bias b^{l+1} are generated. The last output weight b^{l+1} can be analytically calculated in the same way as ELM.

In the first three steps, a hierarchical structure is built, which can extract the high-level features from the input data by an unsupervised manner. In the last step, the learnt features are fed to a basic ELM to figure out the final result. The proposed method still has a fast learning speed, since there is no need to adjust any parameters.

TABLE 1. General activation functions.

Function names	Formulations
Sigmoid	$f(x) = 1/\left(1 + e^{-x}\right)$
Gaussian (Radial basis function)	$f(x) = e^{-x^2}$
Rectified linear unit (ReLu)	$f(x) = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases}$
TanH	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
Hardlim	$f(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$
Sine	$f(x) = \sin(x)$
Triangular basis function	$f(x) = \begin{cases} 1 - \lvert x \rvert, -1 \leq x \leq 1 \\ 0, otherwise \end{cases}$

B. THE SYNTHESIS STRATEGY

According to ELM theory, ELM using a nonlinear piecewise continuous function as the activation function $f(\cdot)$ can approximate any continuous target function Y [27]. Neural network can extract the heterogeneous features by virtue of the nonlinear activation function. However, the neural networks with different activation functions usually show different characteristics and complementary learning behaviors. In order to overcome the limits of the neural network with individual activation function and enhance the generalization performance, this paper uses the ensemble of multiple EDELMs with different activation functions. The general activation functions which suffice for ELM theory are listed in Table 1.

All these functions can be divided into exponential operation and non-exponential operation. Sigmoid function and Gaussian function are the typical exponential operations which are widely used as the activation functions for almost all kinds of neural networks. ReLu and its extension function are developed recently in deep learning field, which outperform the traditional functions in some pattern recognition problems [28], [29]. In order to improve the classification capability, this article employs different activation functions to design different EDELMs. Each EDELM will process the input data independently and offer the classification result.

As the typical classification problem, health state will be estimated by every EDELM, and the estimated results are utilized for deciding the final health states by synthesis strategy. Many researches [30], [31] focus on designing synthesis strategy, among which the majority voting method has been widely used by virtue of its convenience and feasibility. However, the main defect of majority voting strategy is that all the individual models are assigned same weights and treated equally.

In this paper, an improved majority voting strategy is designed to combine all the classification results. The training samples $x = [x_1, x_2, \cdots, x_n]$ are used to train

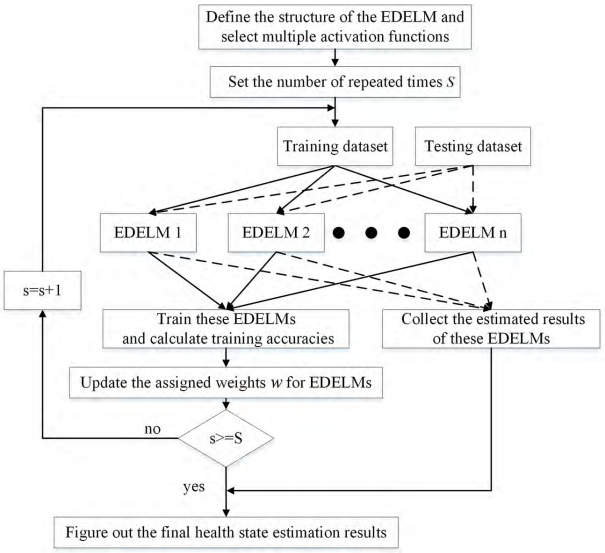


FIGURE 3. The procedure of health state estimation.

these EDELMs, which employ different activation functions. The training results and accuracies $A = [acc^1, acc^2, \cdots, acc^r]$ can be acquired. In order to get reasonable accuracies, the training process is repeated multiple times S and the mean values of the accuracies are used to gain the assigned weights. The superscript r refers to the number of EDELMs. The subscript denotes that the ith sample belongs to the tth class. Then, the weight w^j of the jth individual result can be calculated as follows.

$$w^j = \frac{acc^j}{\sum_{j=1}^{r} acc^j}, \quad \left(\sum_{j=1}^{r} w^j = 1\right). \tag{6}$$

The testing samples are fed into the trained EDELMs for health state estimation. The estimated results $Y_i^r = \left[y_{i1}^r, y_{i2}^r, \cdots, y_{it}^r\right]$ of each EDELM are collected. The weighted results, the ith sample belongs to the tth class in the jth EDELM, are calculated as follows.

$$Y_i^j = \left[y_{i1}^j, y_{i2}^j, \cdots, y_{it}^j\right] w^j. \tag{7}$$

So, the matrix $Y_i = \left[Y_i^1, Y_i^2, \cdots, Y_i^r\right]^T$ is the weighted result confirming that the ith sample belongs to the tth class in all the EDELMs. Then, the belonging of the ith sample is calculated by finding out the maximum among the classes after summing matrix Y_i. The final results of health state estimation can be defined as follows.

$$\hat{Y}_i = \max\left(\sum_{j=1}^{r} \left[y_{i1}^j, y_{i2}^j, \cdots, y_{it}^j\right] w^j\right). \tag{8}$$

C. PROCEDURE FOR HEALTH STATE ESTIMATION

This paper proposes a novel algorithm for health state estimation. The general procedures of health state estimation are summarized as follows and the flowchart is shown in Fig. 3.

TABLE 2. Basic information of the benchmark datasets.

Data sets	No. of real attributes	No. of classes	No. of instances
Ionosphere	34	2	351
Wine	13	3	178
Seeds	7	3	210
Statlog (Heart)	13	2	270
Sonar	60	2	208
Whole	7	2	440

Step 1: Define the health states in accordance with the working principles of the estimation object.

Step 2: Collect the original samples under different health states and divide the original samples into training dataset and testing dataset.

Step 3: Define the structure of the EDELM, select the different activation functions for the EDELMs and set the repeated times S.

Step 4: Use the training dataset to train the EDELMs and figure out the training accuracies.

Step 5: Update the mean value of the training accuracies and calculate the assigned weights.

Step 6: Validate the estimation performance of the trained EDELMs model by testing dataset.

Step 7: Allocate the calculated weights to the estimated results of EDELMs, and report the final estimated results.

IV. EXPERIMENTS AND RESULTS

In this section, the performance of the proposed method is validated firstly. Then, an IMA degradation model is built for collecting the raw data. Finally, the proposed method is applied to estimating the health state. In section 4.1, the proposed method is compared with 8 related methods on the benchmark datasets. In section 4.2, an IMA degradation model is regarded as a stochastic process and built based on intermittent faults. The degradation model is realized on the Simulink platform. In section 4.3, the samples under different health states are acquired and the proposed method is applied to IMA health state estimation. In this paper, all the algorithms are carried out in the Matlab 2013b environment running on a desktop with CPU 2.80GHz and 4GB RAM.

A. BENCHMARKS

Benchmark datasets employed in this study are Ionosphere datasets, Wine datasets, Seed datasets, Statlog datasets, Sonar datasets and Whole datasets. All the benchmark datasets are collected from UCI Machine Learning Repository [32].The number of attributes involved in the classification problem, number of classes and total number of instances of each dataset are listed in Table 2.

This paper integrates three EDELMs with different activation functions to enhance the classification performance. Sigmoid function and Gaussian function are two of the selected activation functions that belong to exponential operation. ReLu function is one of the selected activation functions that belong to non-exponential operation. In order to validate the

TABLE 3. The structures, dropout rates and instances.

Data sets	Structures	Dropout rates	training instances	testing instances
Ionosphere	34-32-16-8-2	0.1	270	81
Wine	13-12-10-8-3	0.2	120	58
Seeds	7-10-12-10-3	0.2	150	60
Heart	13-12-10-8-2	0.2	180	90
Sonar	60-40-36-18-2	0.2	150	58
Whole	7-10-12-10-2	0.1	320	120

TABLE 4. The training results.

		Training Accuracies and Standard Deviations					
		Iono sphere	Wine	Seeds	Heart	Sonar	Whole
Proposed method	Acc	87.13%	97.26%	93.67%	83.67%	86.9%	83.19%
	Std	0.0221	0.0093	0.0085	0.0145	0.0351	0.0133
Method1	Acc	87.52%	97.79%	94.23%	85.37%	90.27%	84.39%
	Std	0.017	0.0136	0.0155	0.0324	0.0284	0.0121
Method2	Acc	81.15%	96.45%	89.76%	82.32%	92.93%	78.26%
	Std	0.0457	0.0209	0.0164	0.0165	0.0227	0.043
Method3	Acc	73.9%	81.07%	65.96%	69.72%	79.67%	81.47%
	Std	0.0408	0.0604	0.0413	0.0411	0.0314	0.0375
Method4	Acc	77.36%	87.72%	87.87%	76.41%	70.39%	80.18%
	Std	0.0502	0.0764	0.0509	0.0433	0.0321	0.0574
Method5	Acc	90.52%	95.31%	91.37%	85.67%	75.71%	70.88%
	Std	0.0537	0.0585	0.0231	0.0244	0.0396	0.0594
Method6	Acc	81.88%	93.17%	83.49%	81.17%	75.8%	83.02%
	Std	0.0394	0.0035	0.038	0.0394	0.0421	0.0381
Method7	Acc	76.67%	83.57%	78.05%	71.58%	70.8%	75.87%
	Std	0.0396	0.0637	0.044	0.0462	0.0345	0.0446
Method8	Acc	79.39%	89.82%	82.01%	78.95%	68.07%	80.6%
	Std	0.0427	0.0493	0.031	0.038	0.0741	0.0333

learning performance, the results of the proposed algorithm are compared with 8 methods. Method 1 is EDELMs without dropout technique; method 2 is EDELM with Sigmoid activation function, method 3 is EDELM with Gaussian activation function; method 4 is EDELM with ReLu activation function; method 5 is SDAE with Sigmoid activation function; method 6 is ELM with Sigmoid activation function; method 7 is ELM with Gaussian activation function; method 8 is ELM with ReLu activation function.

For the purpose of ensuring the learning efficiency of these algorithms, structure of SDAE is the same as that of EDELM algorithm. The number of hidden neurons of ELMs is equal to the number of neurons of the last hidden layer of EDELM. The number of input layer neurons (the output layer neurons) of these algorithms is equal to the number of the input attributions (the target classes). The corruption rate is 0.1. The structures, the dropout rates, the training instances and testing instances of the proposed method for different datasets are listed in Table 3. The pretraining epoch and fine-turning epoch of method 5, SDAE, are set as 200 and 100, respectively.

In this paper, the individual EDELM is repeated 50 times to get reasonable accuracies for calculating the assigned weights. Every algorithm is run 50 times to obtain credible results. The training accuracies (Acc) and the training standard deviations (Std) are listed in Table 4. The testing accuracies and the testing standard deviations are displayed in Table 5.

TABLE 5. The testing results.

		Training Accuracies and Standard Deviations					
		Iono sphere	Wine	Seeds	Heart	Sonar	Whole
Proposed method	Acc	7.59%	96.72%	92.25%	83.39%	81.53%	84.58%
	Std	0.0377	0.0147	0.0197	0.0337	0.0325	0.025
Method1	Acc	85.99%	96.21%	91.83%	81.22%	73.81%	83.62%
	Std	0.0359	0.0249	0.0305	0.0409	0.0452	0.0174
Method2	Acc	84.62%	92.52%	92.08%	83.33%	78.07%	78.05%
	Std	0.0462	0.0264	0.0209	0.0266	0.0528	0.0679
Method3	Acc	75.16%	76.21%	65.7%	68.37%	62.37%	84.22%
	Std	0.0572	0.0736	0.0637	0.0571	0.0647	0.0539
Method4	Acc	80.67%	85.93%	90.13%	78.78%	73.08%	81.2%
	Std	0.0652	0.0839	0.071	0.0615	0.0592	0.0866
Method5	Acc	87.53%	93.64%	90.25%	80.94%	81.29%	71.73%
	Std	0.0487	0.0695	0.0402	0.0355	0.0428	0.0648
Method6	Acc	80.42%	89.72%	82.52%	78.78%	71.24%	82%
	Std	0.0462	0.0619	0.0572	0.0457	0.0575	0.0541
Method7	Acc	74.44%	78.83%	76%	67.64%	61.97%	74.49%
	Std	0.0532	0.0962	0.0592	0.0628	0.07	0.061
Method8	Acc	77.8%	86.97%	80.1%	75.6%	75.76%	79.44%
	Std	0.0587	0.0664	0.0511	0.0574	0.041	0.0476

TABLE 6. The training time and testing time.

	Training time /testing time					
	Iono sphere	Wine	Seeds	Heart	Sonar	Whole
Proposed method	24.5/5.89	6.7/1.5	7.3/1.8	8.4/2.4	27.4/4.02	14.5/1.91
Method1	24.2/5.63	6.7/1.6	7.1/1.7	8.6/2.3	27.5/3.92	14.4/1.88
Method2	8.4/1.7	2/0.43	2.2/0.44	2.4/0.51	8.6/1.1	3.9/0.52
Method3	8.9/2.1	2.4/0.5	2.4/0.56	2.8/0.95	9.6/1.3	4.2/0.63
Method4	4.9/0.67	1.6/0.36	1.4/0.31	1.7/0.42	5.2/0.71	3.3/0.39
Method5	9541/4.3	7425/2.6	7369/2.5	7555/2.7	9464/4.8	8573/2.8
Method6	0.29/0.15	0.13/0.11	0.18/0.11	0.13/0.13	0.48/0.45	0.22/0.14
Method7	0.36/0.31	0.16/0.14	0.19/0.15	0.18/0.17	0.47/0.34	0.29/0.18
Method8	0.178/0.1	0.08/0.09	0.11/0.09	0.08/0.09	0.29/0.23	0.14/0.1

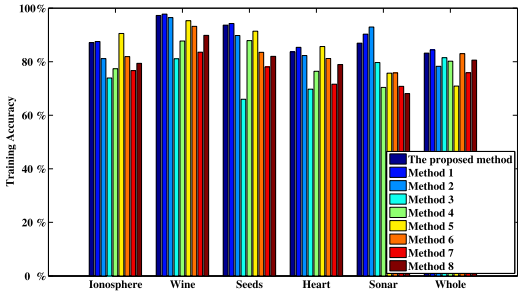


FIGURE 4. The training accuracies.

The training time and the testing time (in milliseconds) of the algorithms on these datasets are listed in Table 6.

To make it clear, Fig. 4 shows the training accuracies on different datasets. Fig. 5 shows the testing accuracies on different datasets. The training time and testing time are displayed in Fig. 6 and Fig. 7, respectively.

As the above tables and figures show, the proposed method has small standard deviations in both training step and testing step due to that ELM autoencoder and dropout technique are adopted to determine the stable and robust weights. Although some algorithms have better training performance in some cases, the proposed method still has a stable and good training

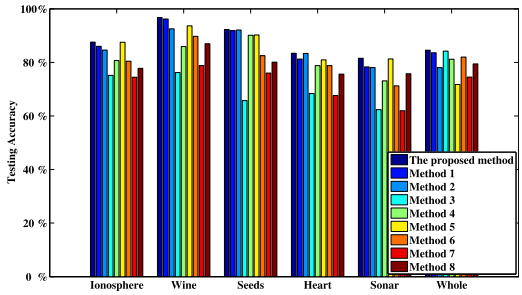


FIGURE 5. The testing accuracies.

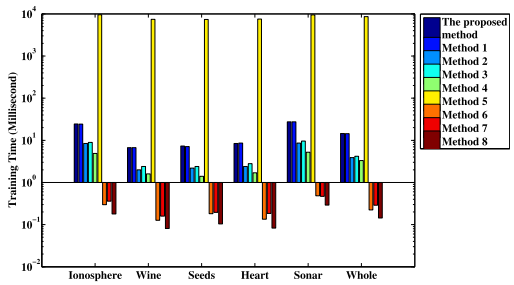


FIGURE 6. The training time.

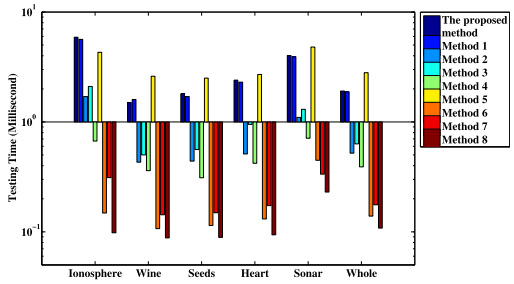


FIGURE 7. The testing time.

capability for different datasets. Compared with method 2, method 3 and method 4, the ensemble manner provides a better generalization performance for the proposed method. Compared with method 6, method 7 and method 8, the proposed method shows the best classification capability due to the excellent performance of deep learning in feature extraction. The testing results further confirm that the proposed method outperforms other algorithms. The proposed method not only has outstanding classification ability and stable performance, but also adapts to all kinds of datasets. Although the SDAE has an excellent testing performance for some datasets, the training time of SDAE is the longest among all these algorithms. On the contrary, the ELM theory enables the proposed method to keep the fast learning speed.

B. IMA DEGRADATION MODEL

This paper focuses on the health state estimation of IMA, and therefore an IMA degradation model is built. IMA is a generalized and modular hardware platform, where the system functions are realized as segregated software blocks [33].

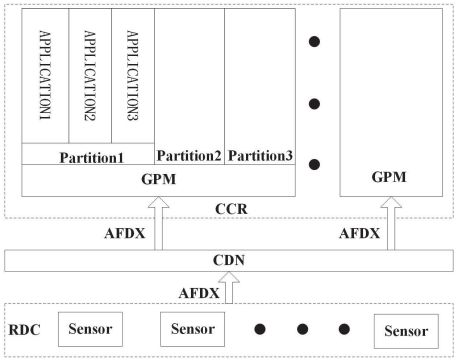


FIGURE 8. IMA structure.

As a core system of modern airplanes, a scheduling is set up to allocate the time slices to all the functions. The time slice is a little longer than the function execution time. To satisfy the real-time requirement, all the functions should be completed in the allocated time slice. During the fly cycles, all the functions will be repeatedly executed according to the defined scheduling. IMA consists of a number of line replaceable modules, mainly including remote data concentrator (RDC), common data network (CDN) and common computing resource (CCR). Generally, AFDX is used as data bus standard for CDN, general processor module (GPM) is the main hardware for function realizing in CCR. The IMA structure is shown in Fig. 8.

In modern airplane, a health monitoring function is hosted on IMA to enhance the reliability and safety. The health monitoring function aims to handle the faults which may occur during the fly cycle. If the fault can be fixed in time and the function is still working in a normal condition, the fault is defined as the intermittent fault. IMA will record the messages about the intermittent faults (IFs). In this paper, IFs are regarded as the feature of IMA.

There are many reasons responsible for IFs, for instance, extreme environment, extreme vibration and electronic devices degradation. In the field of civil aviation, extreme working condition is rarely encountered. However, the electronic devices degradation cannot be neglected. Many research works have presented that almost all the reasons for IFs, such as time-dependent dielectric breakdown (TDDB), hot carrier injection (HCI) and electromigration (EM), are caused by wearout resulting from extensive use [34]. IFs can happen in a low frequency at the initial stage, which are identified as small noise fluctuation. Afterwards, the frequency and duration increase, IFs start to occur. Finally, IFs will seriously impact on the system, even can lead to a failure [35]. As a real-time system, IMA must complete the functions in a limited time. If there is an intermittent fault, the health monitor function will handle it. Meanwhile, the handling process will occupy the stipulated time of the error function. If the function cannot complete in the allocated time due to the frequent IFs, this condition is defined as failure.

In this paper, an IMA degradation model is built based on

IFs. As mentioned above, every basic function in IMA will be assigned a fixed time slice for completing the operation. The function completion time $X(t)$ consists of original function execution time and the time delay caused by error handling. Suppose that the performance of IMA is constant in a fly cycle, the function completion time $X(t)$ can be proved that it has the following properties in the allocated time slice. Firstly, it owns stationary increments and independent increments. Secondly, the initial time $X(0) = 0$. Thus, the practical execution time can be depicted using LÃĺvy process. The general LÃĺvy process can be formulated as follows [36].

$$X(t) = \mu t + B(t) + \int_{[0,t]\times\{|x|>1\}} xM(ds, dx) + \int_{[0,t]\times\{|x|\le 1\}} x(M - m)(ds, dx). \quad (9)$$

where μ is called drift parameter, $B(t)$ is Wiener process. $\int_{[0,t]\times\{|x|>1\}} xM(ds, dx)$ is a Poisson process, where M is Poisson stochastic measure in $R^+ \times R_0$, denoting the frequency of jump with scale $[s, s+ds]$ in the time interval $[x, x+dx]$; $\int_{[0,t]\times\{|x|\le 1\}} x(M-m)(ds, dx)$ is a pure jump martingale, where $m(ds, dx)$ is mean measure of M, which satisfies $\int (x^2 \wedge 1) v(dx) < \infty$.

In view of the IMA, the computing power scarcely changes. Hence, pure jump martingale and drift process can be discarded. In this paper, the function completion time $X(t)$ is defined as follows.

$$X(t) = \mu t + B(t) + \sum_{j=0}^{N(t)} W_j, \quad N(t) = 0, 1, 2 \ldots. \quad (10)$$

where $\mu t + B(t)$ denotes the function execution time without IFs; $\sum_{j=0}^{N(t)} W_j$ is a compound Poisson process. W_j is the time delay caused by the jth error handling; $N(t)$ denotes the total number of IFs by time t.

As mentioned above, TDDB, HCI and EM which should be considered in IMA are the typical degradation reasons for electronic devices. Many researches devote to revealing the mechanism of TDDB, HCI and EM. The significant conclusions are summarized in the light of lots of experimental results. The degradation of electronics caused by TDDB and HCI can be described by Weibull distribution [37]. The degradation of electronics caused by EM can be described by Lognormal distribution [38]. The reliability of the electronic device in IMA which is effected by single factor can be calculated as follows.

$$R(t) = P(X > t) = \int_t^{\infty} f(x)dx. \quad (11)$$

where $f(x)$ is probability density function. Hence, multiple factors have multiple probability density functions $f_1(x) f_2(x) \cdots f_n(x)$ which correspond to multiple reliabilities $R_1(t), R_2(t), \cdots, R_n(t)$. As reliability denotes the whole life of IMA, a fly cycle, which is a certain time interval, is used to discretize it in this paper. Therefore, in the certain

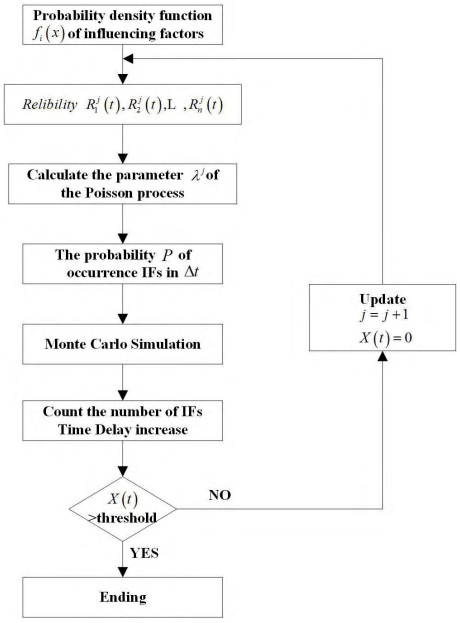


FIGURE 9. The simulation flowchart.

time interval, a correlation is obliged to exist between $R_i^j(t)$ and the parameter $\lambda_i^j(t)$ of Poisson process $N_i^j(t)$. The superscript and subscript are the jth time interval and the ith factor, respectively. The relationship between $R_i^j(t)$ and parameter $\lambda_i^j(t)$ can be defined as follows.

$$\lambda_i^j(t) = C \cdot \sqrt{\frac{1}{R_i^j(t)}}. \quad (12)$$

where C is coefficient. The occurrence of IFs in a certain time interval can be regarded as a homogeneous Poisson process, although the frequency of IFs is changing over time. It can be proved that the occurrence of IFs with multiple factors follows homogeneous Poisson process and its parameter $\lambda^j(t) = \sum_i^n \lambda_i^j(t)$ can be calculated based on the additivity of Poisson process.

The parameters of the Wiener process and probability density function should be determined, besides, the IMA degradation model should initialize the time interval and threshold. This paper adopts Monte Carlo method to simulate the IMA degradation process. Fig. 9 shows the simulation flowchart.

C. IMA HEALTH STATE ESTIMATION

In this paper, the left fifth GPM of Boeing 787 is simulated. The left fifth GPM of IMA hosts flight management function (FMF), thrust management function (TMF) and navigation display (ND). The simulation is realized by Simulink. As Fig. 10 shows, RDC provides the flight data for GPM via AFDX. A scheduling is defined to control execution sequence of the hosted functions. As the core function, FMF uses the navigation data base and performance data base to calculate the flight data, and sends the calculated datasets to TMF and ND. During the function execution time of FMF, the simu-

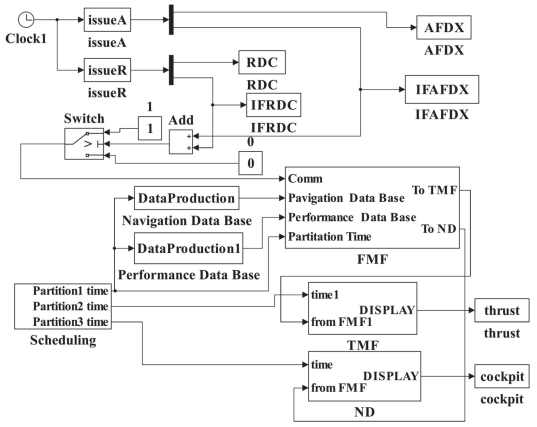


FIGURE 10. The simulation model.

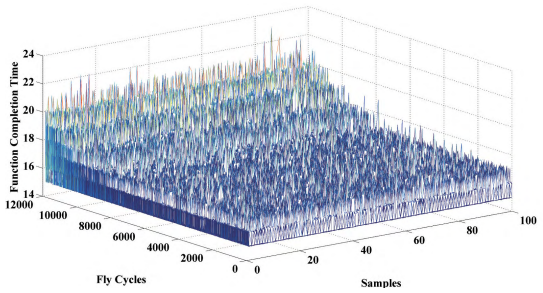


FIGURE 11. The function completion time.

lation model will generate IFs according to Figure 9. In this simulation, the period of the scheduling is set as 50 milliseconds. The execution time of FMF is around 15 milliseconds and the allocated time slice is set as 20 milliseconds. The time slices for TMF and ND are 20 milliseconds and 10 milliseconds, respectively. The simulation consists of 12000 fly cycles and the scheduling will be repeated 100 times in a fly cycle. Fig. 11 shows the simulated data of IMA degradation process.

Generally, the whole life of a system can be divided into four health states which are defined as fully functionality, degraded functionality, reduced functionality and no functionality. The simulation data are divided into 4 parts depending on the frequency of the IFs. The first 10862 fly cycles of simulation data are defined as fully functionality. The fly cycles from 10863 to 11370 are defined as degraded functionality. The fly cycles from 11370 to 11727 are defined as reduced functionality. The last 273 fly cycles are defined as no functionality.

This paper acquires the last 2000 fly cycles as the raw data for health state estimation. The raw data are divided into 1600 sets of training samples and 400 sets of testing samples. The pertaining epoch and fine-turning epoch of SDAE are set as 1000 and 500, respectively. The corruption rates, the dropout rates and the structures of these algorithms are listed in Table 7.

All the evaluation processes are repeated 50 times. The training accuracies, training standard deviations, training time (in milliseconds), testing accuracies, testing standard deviations and testing time (in milliseconds) are presented in Table 8.

Fig. 12 shows the training and testing results of IMA health state estimation. Fig. 13 indicates the training time and testing time for evaluating the health states of IMA.

As Table 8 and Fig. 12 show, the proposed method has an outstanding training capability and the best testing accuracy. Meanwhile, the proposed method has small standard deviation, which illustrates that the proposed method possesses a stable performance. Fig. 13 indicates that the classical SDEA is time consuming to train the estimation model, which can

TABLE 7. Corruption rates, dropout rates and structures.

	Corruption rates	Dropout rates	Structures
Proposed method	0.2	0.2	100-200-100-200-4
Method 1,2,3,4	/	/	100-200-100-200-4
Method 5 (SDAE)	0.2	0.2	100-50-20-10-4
Method 6,7,8	/	/	100-200-41

TABLE 8. Training and testing results.

	Training Acc	Training Std	Training Time	Testing Acc	Testing Std	Testing Time
Proposed method	96.23%	0.0289	2024	81.1%	0.0196	51.7
Method1	91.51%	0.0164	2003	76.14%	0.0221	49.5
Method2	90.06%	0.0103	83.9	72.75%	0.0225	25.6
Method3	75.11%	0.0272	762	66.78%	0.0402	26.3
Method4	84.62%	0.0168	673	69.5%	0.0238	12.6
Method5	88.01%	0.0269	107033	79.19%	0.0281	6
Method6	81.85%	0.0086	173	69.54%	0.0211	8.5
Method7	77.86%	0.0101	183	65.78%	0.0221	13.1
Method8	79.37%	0.0087	141	68.36%	0.0255	4.8

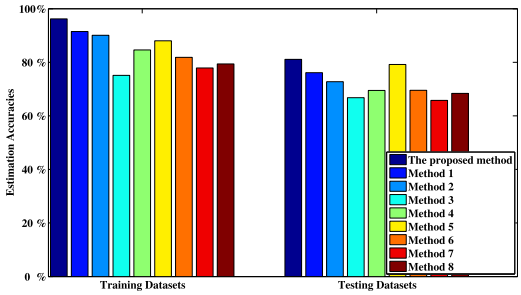


FIGURE 12. The estimation results.

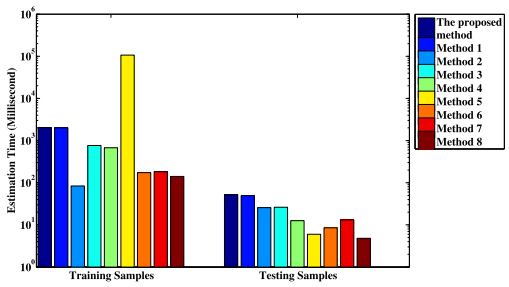


FIGURE 13. The estimation time.

not suffice for real-time system. Different from the conventional back propagation method, the proposed method needs not to adjust the parameters, which makes it take an acceptable time. The basic ELM takes less time than the proposed method, nevertheless, its estimation performance is unstable and unattractive. As all the results show, the proposed method can learn hierarchical features from the raw data immediately, and retain the reliable information based on the deep structure. ELM has the excellent performance and fast learning speed while working as a classifier or feature extractor. All the advantages of the proposed method analyzed from the results confirm that it is more suitable for health state estimation of IMA.

V. CONCLUSION

This paper focuses on the health state estimation of IMA. To enhance the safety and satisfy the real-time requirement of IMA, a novel deep learning method with fast learning speed called EDELM is proposed for health state estimation. The proposed method inherits the advantages of deep learning and ELM. Deep learning can learn the high-level features from the raw data directly, and ELM algorithm can guarantee a fast learning speed for the deep structure. Firstly, EDELM applies ELM autoencoder and dropout technique to determine weights between the adjacent layers for further learning the deep features. Then, EDELM uses the learnt features to figure out the estimation results by a basic ELM. Finally, this paper takes advantage of the ensemble of multiple EDELMs with different activation functions to improve the generalization performance. A synthesis strategy is designed for combining the estimation results of the EDELMs.

The performance of the proposed method is compared with the existing methods on the benchmark datasets firstly. Secondly, the IMA degradation is regarded as a stochastic process. The degradation model is built based on the intermittent faults and the simulation datasets under different health states are collected. Thirdly, the proposed method is applied to IMA health state estimation. The results confirm that the proposed method can capture the features automatically and retain more useful information than manual feature extraction. Compared with other methods, the proposed method is more robust and more accurate. Since the proposed method adopts the theory of ELM, it spends a little time for health state estimation, which enables the proposed method to suit to real-time system. In the future, the research should focus on the remaining useful life prediction.

REFERENCES

[1] H. Wang, T. Zhao, F. Ren, and Z. Jiang, "Integrated modular avionics system safety analysis based on model checking," in *Proc. Annu. Rel. Maintainability Symp.*, Jan. 2017, pp. 1–6.

[2] X. M. Liu, C. Cao, X. Zhao, J. Sun, and J. Zhu, "Network performance analysis of time-triggered Ethernet based on network calculus for DIMA," in *Proc. IEEE/AIAA 34th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2015, pp. 10A3-1–10A3-7.

[3] M. Halle and F. Thielecke, "Model-based transition of IMA architecture into configuration data," in *Proc. IEEE/AIAA 35th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2016, pp. 1–10.

[4] Z. Li, S. Wang, T. Zhao, and B. Liu, "A hazard analysis via an improved timed colored Petri net with time–space coupling safety constraint," *Chin. J. Aeronaut.*, vol. 29, no. 4, pp. 1027–1041, 2016.

[5] P. Tamilselvan and P. Wang, "Failure diagnosis using deep belief learning based health state classification," *Rel. Eng. Syst. Saf.*, vol. 115, pp. 124–135, Jul. 2013.

[6] Z. Liu, M. J. Zuo, and Y. Qin, "Remaining useful life prediction of rolling element bearings based on health state assessment," *Proc. Inst. Mech. Eng., C, J. Mech. Eng. Sci.*, vol. 230, no. 2, pp. 314–330, 2016.

[7] F. Yang, M. S. Habibullah, T. Zhang, Z. Xu, P. Lim, and S. Nadarajan, "Health index-based prognostics for remaining useful life predictions in electrical machines," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2633–2644, Apr. 2016.

[8] Z. Gao, C. Ma, D. Song, and Y. Liu, "Deep quantum inspired neural network with application to aircraft fuel system fault diagnosis," *Neurocomputing*, vol. 238, no. 17, pp. 13–23, 2017.

[9] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[10] Z. Liu, J. Zhen, C.-M. Vong, S. Bu, J. Han, and X. Tang, "Capturing high-discriminative fault features for electronics-rich analog system via deep learning," *IEEE Trans. Ind. Inform.*, vol. 13, no. 3, pp. 1213–1226, Jun. 2016.

[11] C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," *Signal Process.*, vol. 130, pp. 377–388, Jan. 2017.

[12] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Rel. Eng. Syst. Saf.*, vol. 172, pp. 1–11, Apr. 2018.

[13] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 1, nos. 1–3, pp. 489–501, 2006.

[14] Y. Peng, W. Kong, and B. Yang, "Orthogonal extreme learning machine for image classification," *Neurocomputing*, vol. 266, pp. 458–464, Nov. 2017.

[15] G.-B. Huang, M.-B. Li, L. Chen, and C.-K. Siew, "Incremental extreme learning machine with fully complex hidden nodes," *Neurocomputing*, vol. 71, nos. 4–6, pp. 576–583, 2008.

[16] H.-J. Rong, Y.-S. Ong, A.-H. Tan, and Z. Zhu, "A fast pruned-extreme learning machine for classification problem," *Neurocomputing*, vol. 72, no. 1, pp. 359–366, 2008.

[17] S. Scardapane, D. Comminiello, M. Scarpiniti, and A. Uncini, "Online sequential extreme learning machine with kernels," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2214–2220, Sep. 2015.

[18] Y. Wang, F. Cao, and Y. Yuan, "A study on effectiveness of extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2483–2490, 2011.

[19] M. Luo, C. Li, X. Zhang, R. Li, and X. An, "Compound feature selection and parameter optimization of ELM for fault diagnosis of rolling element bearings," *ISA Trans.*, vol. 65, pp. 556–566, Nov. 2016.

[20] R. Rajesh and J. S. Prakash, "Extreme learning machines—A review and state-of-the-art," *Int. J. Wisdom Based Comput.*, vol. 1, no. 1, pp. 35–49, 2011.

[21] H. Zhang, S. Zhang, and Y. Yin, "Online sequential ELM algorithm with forgetting factor for real applications," *Neurocomputing*, vol. 261, pp. 144–152, Oct. 2017.

[22] L. L. C. Kasun, H. Zhou, G.-B. Huang, and C. M. Vongc, "Representational learning with ELMs for big data," *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 31–34, Dec. 2016.

[23] G. Huang, Z. Bai, L. L. C. Kasun, and C. M. Vong, "Local receptive fields based extreme learning machine," *IEEE Comput. Intell. Mag.*, vol. 10, no. 2, pp. 18–29, May 2015.

[24] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006.

[25] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 809–821, Apr. 2017.

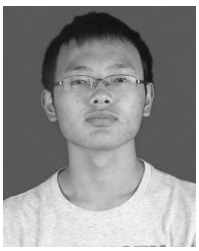
[26] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, 2010.

[27] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

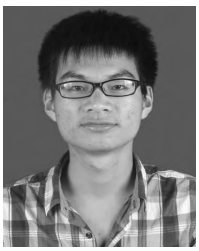
[28] D. Yarotsky, "Error bounds for approximations with deep ReLU networks," *Neural Netw.*, vol. 94, pp. 103–114, Oct. 2017.
[29] G. Lee, Y. W. Tai, and J. Kim, "ELD-Net: An efficient deep learning architecture for accurate saliency detection,"İ *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, pp. 1599–1610, 2018.
[30] A. A. Silva, S. Gupta, A. M. Bazzi, and A. Ulatowski, "Wavelet-based information filtering for fault diagnosis of electric drive systems in electric ships," *ISA Trans.*, vol. 78, pp. 105–115, Jul. 2018.
[31] T. E. Stone and S. R. McKay, "Majority-vote model on a dynamic small-world network," *Phys. A, Stat. Mech. Appl.*, vol. 419, pp. 437–443, Feb. 2015.
[32] X. Liu, Z. Liu, G. Wang, Z. Cai, and H. Zhang, "Ensemble transfer learning algorithm," *IEEE Access*, vol. 6, pp. 2389–2396, 2018.
[33] Z. Gao, C. Ma, Y. Luo, and Z. Liu, "IMA health state evaluation using deep feature learning with quantum neural network," *Eng. Appl. Artif. Intell.*, vol. 76, pp. 119–129, Nov. 2018.
[34] M. Demertzi, B. Zandian, R. Rojas, and M. Annavaram, "Benchmarking ISA reliability to intermittent errors," in *Proc. IEEE Int. Symp. Workload Characterization*, Nov. 2012, pp. 86–87.
[35] A. Correcher, E. Garcia, F. Morant, E. Quiles, and L. Rodriguez, "Intermittent failure dynamics characterization," *IEEE Trans. Rel.*, vol. 61, no. 3, pp. 649–658, Sep. 2012.
[36] Z. Li and W. Xu, "Asymptotic results for exponential functionals of Lévy processes," *Stochastic Process. Appl.*, vol. 128, no. 1, pp. 108–131, 2018.
[37] B. Cai, Y. Liu, and M. Xie, "A dynamic-Bayesian-network-based fault diagnosis methodology considering transient and intermittent faults," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 276–285, Jan. 2017.
[38] M. Pecht and V. Ramappan, "Are components still the major problem: A review of electronic system and device field failure returns," *IEEE Trans. Compon., Hybrids, Manuf. Technol.*, vol. 15, no. 6, pp. 1160–1164, Dec. 1992.

ZEHAI GAO received the master's degree from Northwestern Polytechnical University, China, in 2014, where he is currently pursuing the Ph.D. degree with the School of Aeronautics. His research interests include pattern recognition and fault diagnosis.

CUNBAO MA received the B.S., master's, and Ph.D. degrees from Northwestern Polytechnical University (NWPU), China, in 1984, 1987, and 2011, respectively. He is currently a Professor with the School of Aeronautics, NWPU. His research interests include flight simulation and system modeling.



ZHIYU SHE received the B.S. degree from the School of Aeronautics, Northwestern Polytechnical University, Xi'an, China, in 2017, where he is currently pursuing the Ph.D. degree. His main research areas include flight safety analysis.



XU DONG received the B.S. degree from the School of Aeronautics, Northwestern Polytechnical University, Xi'an, China, in 2014, where he is currently pursuing the master's degree. His main research areas include system modeling and health assessment of integrated modular avionics.